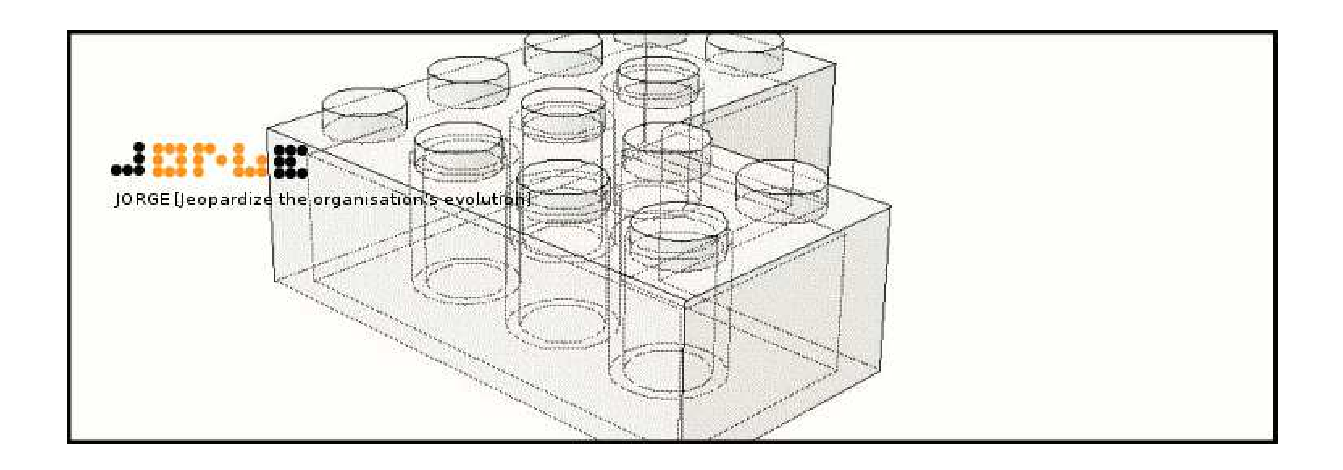



Diplomarbeit
LEGO Mindstorms Simulator - JORGE
Benutzerhandbuch

christof.seiler@datacomm.ch
nik@netstyle.ch
stefan.feissli@hispeed.ch

16. Dezember 2005

# Was ist JORGE?

JORGE ist eine 3D Simulation für LEGO Mindstorms Roboter. Wie beim Vorbild aus der realen Welt, kann man ein Programm auf einen (virtuellen) Roboter laden, das Programm starten und das Verhalten des Roboters beobachten. Zur Darstellung des Roboters und der ihn umgebenden Landschaft nutzt JORGE eine dreidimensionale Ansicht, wie man es von aktuellen Spielen kennt. Der grösste Vorteil der Simulation besteht in der Nachbildung von physikalischen Eigenschaften: JORGE bietet zum Beispiel die Möglichkeit an, die Gravitation der virtuellen Welt zu verändern oder die Kraft der Motoren zu erhöhen.

Zum einen dient dieses Dokument zur kurzen Einführung und Erklärung der wichtigsten Funktionen von JORGE. Zum anderen richtet es sich an Anwender die interessiert sind, Roboter Programme in Java zu schreiben und sie zu testen. Es setzt voraus, dass JORGE erfolgreich installiert wurde. Hierzu stellen wir eine ausführliche *Installationsanleitung* zur Verfügung.

# Inhaltsverzeichnis

# Kapitel 1

# Erste Schritte

In diesem Kapitel wird Schritt für Schritt erklärt, wie Sie Hindernisse und Roboter in die virtuelle Welt einfügen können. Im zweiten Teil wird ein Roboter-Programme erstellt und auf dem virtuellen Roboter ausgeführt.

## 1.1 Der Aufruf von JORGE

Wie unter Windows üblich, kann JORGE bequem über den entsprechenden Eintrag im Start-Menü gestartet werden. Wenn Sie bereits eine JORGE-Projektdatei haben, wird JORGE ausgeführt wenn Sie diese Datei öffnen (Doppelklick).

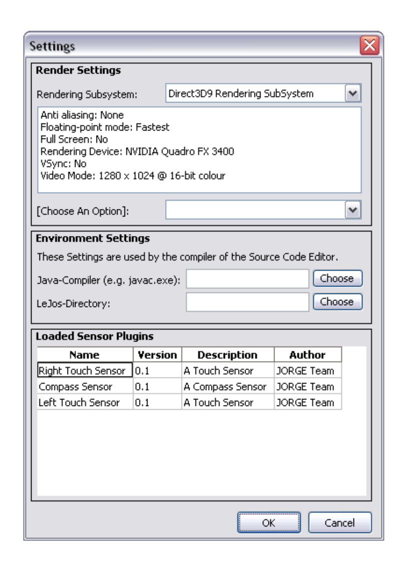



Abbildung 1.1: Einstellungen

Wenn Sie JORGE das erste Mal starten erscheint der obige Dialog, damit können Sie die wichtigsten Einstellungen vornehmen:

- *Render Settings:* Hier könnten Sie das Darstellungssystem für 3D auswählen. OpenGL und Direct3D7 werden leider nicht unterstützt.
- *Environment Settings:* Um Programme mit dem in JORGE eingebauten Editor (dazu später mehr) zu schreiben, muss angegeben werden, wo das Programm *javac.exe* zu finden ist, und wo Sie LeJOS installiert haben. Klicken Sie auf die Schaltfläche *Choose*, um die Einstellungen vorzunehmen. Ein Beispiel:

– *Java-Compiler:* C:\j2sdk1.4.2_09\bin\javac.exe

– *LeJos-Directory:* C:\java\lejos\

Eine kurze Anleitung zur Installation von LeJOS und Java finden Sie im Anhang A bzw. B.

- *Loaded Sensor Plugins:* Hier wird lediglich angezeigt, welche Sensoren geladen wurden.

Nachdem Sie alle Einstellungen vorgenommen haben, können Sie die Konfiguration mit OK abschliessen.

## 1.2 JORGE gestartet - Was nun?

Nun zeigt JORGE ein leeres Fenster mit deaktivierten Schaltflächen an. Dies liegt daran, dass Sie noch

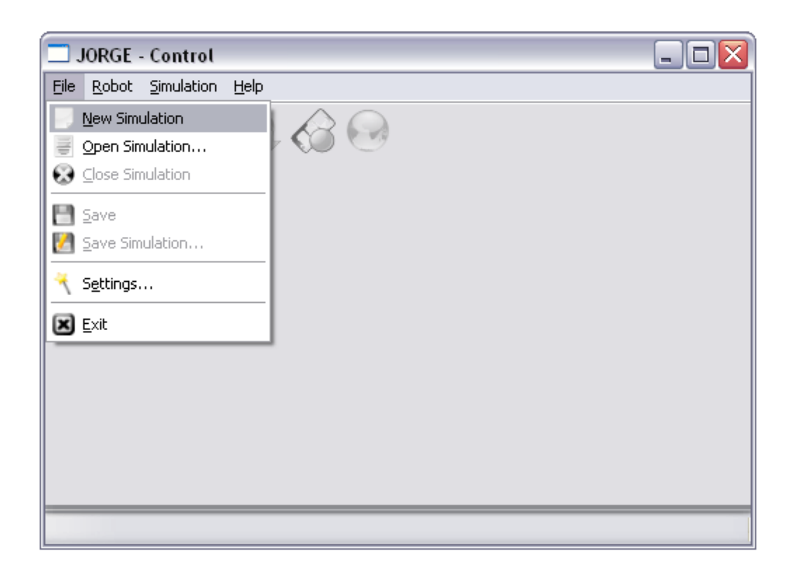



Abbildung 1.2: Haupt Fenster

keine Simulation erstellt bzw. geöffnet haben. Eine Simulation können Sie mit einem Dokument in einem Textverarbeitungsprogramm vergleichen: In einer Simulation werden sämtliche Einstellungen gespeichert. Dies ermöglicht es Ihnen, eine Simulation vorzubereiten und verschiedene Roboterprogramme mit genau gleichen Voraussetzungen zu testen.

Wählen Sie, wie in der Abbildung 1.2 abgebildet, die Funktion File ▷ New Simulation, um eine neue Simulation anzulegen. Im Abschnitt 2.2 wird auf den nun angezeigten Dialog, in dem es darum geht die Welt auszuwählen und die Gravitation für diese Welt festzulegen, genauer eingegangen. Wählen Sie *Done,* um mit den Standardeinstellungen fortzufahren.

## 1.3 Hauptfenster - Welt Konfigurations Modus

Jetzt wird eine neue Simulation angelegt, und die Benutzeroberfläche verändert ihr Aussehen: Sie befinden sich nun im *"Welt Konfigrations Modus".* In diesem Modus geht es darum, Roboter und Hindernisse in der Landschaft zu platzieren.

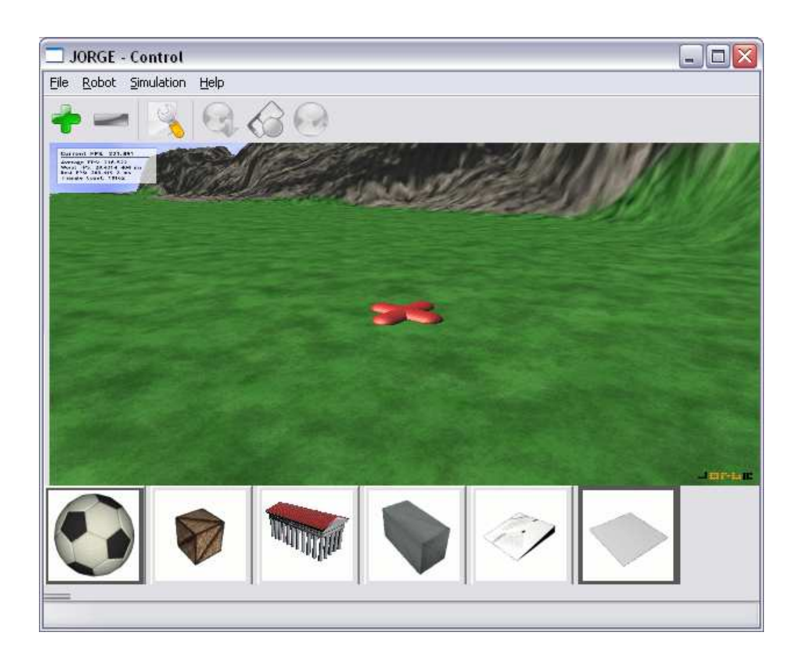



Abbildung 1.3: Hauptfenster

Auf die einzelnen Bestandteile der Benutzeroberfläche wird im Abschnitt 2.1 genauer eingegangen. Sie können nun zu einer virtuellen Erkundungsreise aufbrechen und sich die Welt ansehen. Die Navigation erfolgt mit Maus und Tastatur:

- Mit der Maus verändern Sie die Blickrichtung.
- Mit den Pfeiltasten der Tastatur können Sie die Kamera in die entsprechende Richtung verschieben. Die Tasten *Page Up* und *Page Down* bewegen die Kamera horizontal.
- Mit der Taste Q, können Sie den Anzeige-Modus verändern:
  1. Standard Ansicht: Die Landschaft wird mit allen Details angezeigt.
  2. Gitternetz Ansicht: Die Oberfläche wird nur mit Linienansicht dargestellt.
  3. Punkte Ansicht: Die Oberfläche wird nur durch Punkte repräsentiert.

### 1.3.1 Hindernisse Einfügen

Wenn Sie mit der Maus im unteren Bereich des Fensters auf den Fussball klicken, wird ein Fussball in die Simulation eingefügt. Das rote Kreuz in der Mitte zeigt an wo genau das gewählte Hindernis eingefügt wird.

Mit Hindernissen können Sie einen Parcours für Ihre Roboter gestalten. Im Abschnitt 3.1 wird erklärt wie Sie selber hergestellte Hindernisse hinzufügen können.

### 1.3.2 Roboter Einfügen

Die Roboter sind der wichtigste Teil der Simulation. Über das Menü Robot ▷ Add Robot wird ein neuer Roboter in die Simulation eingefügt. Alternativ können Sie auch das grüne Kreuz in der Toolbar zum Einfügen von Robotern benutzen.

Wie die Hindernisse, werden auch Roboter an der Stelle eingefügt, wo das rote Kreuz angezeigt wird. Die Abbildung 1.4 zeigt JORGE nachdem ein Roboter eingefügt wurde. Für jeden eingefügten Roboter wird auf der linken Seite ein Reiter erstellt, *RobotTab* genannt. Über Ihn können alle Einstellungen die einen Roboter betreffen, vorgenommen werden. Der Roboter, der zum aktuellen *RobotTab* gehört, wird durch einen roten Kreis markiert.




Abbildung 1.4: JORGE mit einem Roboter

## 1.4 Ein bestehendes Programm Laden

Ein LeJOS Programm kann über die Schaltfläche *Load* auf dem *RobotTab* ausgewählt werden. Sobald das Programm geladen ist, wird dieses innerhalb des *RobotTabs* im Textfeld *Robot Log* angezeigt. Wenn Sie JORGE nur testen wollen, können Sie Beispielprogramme laden, welche Sie unter *<JORGE Installations-Ordner>/lejos/* finden. Wie man eigenen Programme schreiben und laden kann, wird im nächsten Kapitel beschrieben.

## 1.5 Editor - Ein Programm erstellen

### 1.5.1 Programm Erstellen

Wählen Sie nun *Edit Program Source*: Ein Fenster mit dem Namen *Source Code Editor* wird geöffnet. Wählen Sie die Aktion File ▷ New... um eine neue Datei zu erstellen. Jetzt muss ein Verzeichnis und ein Dateiname für die neue Datei angegeben werden. Für dieses Beispiel wählen wir den Dateinamen *HelloRobot.java*. Bevor Sie nun mit dem Schreiben des Programms beginnen können, müssen Sie eine weitere Frage von JORGE beantworten: Soll die Datei die Sie eben erstellt haben, ins Projekt aufgenommen werden? Wenn Sie die Frage mit *Ja* beantworten, wird das nächste Mal, wenn Sie auf die Schaltfläche *Edit Program Source* klicken, die Datei *HelloRobot.java* geöffnet. Wenn Sie dies nicht möchten, antworten Sie mit Nein.

Sie können nun die folgenden Zeilen Quelltext in den Editor kopieren oder selber ein Programm schreiben. Wenn Sie mehr Informationen zu den eingesetzten Funktionen erhalten möchten, können Sie in der Toolbar auf die Schaltfläche *Open LeJos Api* drücken (in ihrem Browser wird dann die LeJOS Dokumentation geöffnet).

```
import josx.platform.rcx.*;
public class HelloRobot
{
    public static void main(String args[]) throws Exception
    {
        Motor.A.forward();
        Motor.C.forward();
```

```
            Thread.sleep(2000);

            Motor.C.stop();
            Thread.sleep(2000);
            Motor.C.stop();
            Motor.A.stop();

            Motor.A.backward();
            Motor.C.forward();
            Thread.sleep(6000);

            Motor.A.stop();
            Motor.C.stop();
        }
    }
```

### 1.5.2 Das Programm Kompilieren, Linken und Laden

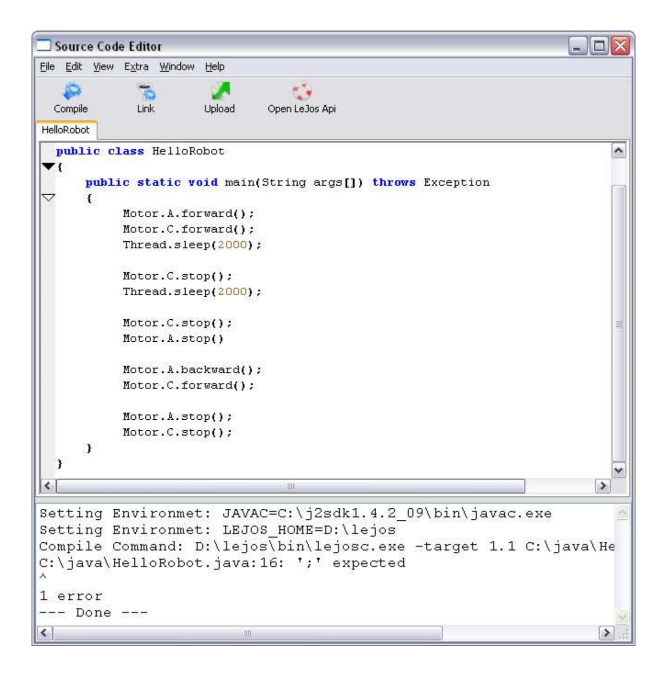



Abbildung 1.5: Kompilieren

Abbildung 1.5 zeigt den Editor, nachdem die Aktion *Compile* ausgeführt wurde. Im unteren Teil ist die Ausgabe des Compilers zu sehen, der einen Fehler beim kompilieren festgestellt hat. Wenn der Compiler keine Fehler beanstandet, kann das Programm mit der Aktion *Link* gelinkt werden, und wenn auch diese Aktion erfolgreich ist, kann das Programm mit der Schaltfläche *Upload* auf den Roboter übertragen werden.

## 1.6 Hauptfenster - Roboter Ansehen Modus

Eigentlich ist nun alles bereit um die Simulation und somit das Programm zu starten. Der zweite Modus ermöglicht es Ihnen, den Roboter von allen Seiten anzusehen und die Farbe zu verändern. Um in diesen Modus zu wechseln, können Sie die Aktion Robot ▷ View Robot im Menü auslösen oder das entsprechende Symbol in der Toolbar wählen. Die Abbildung 1.6 zeigt, den Roboter nachdem die Farbe verändert wurde.

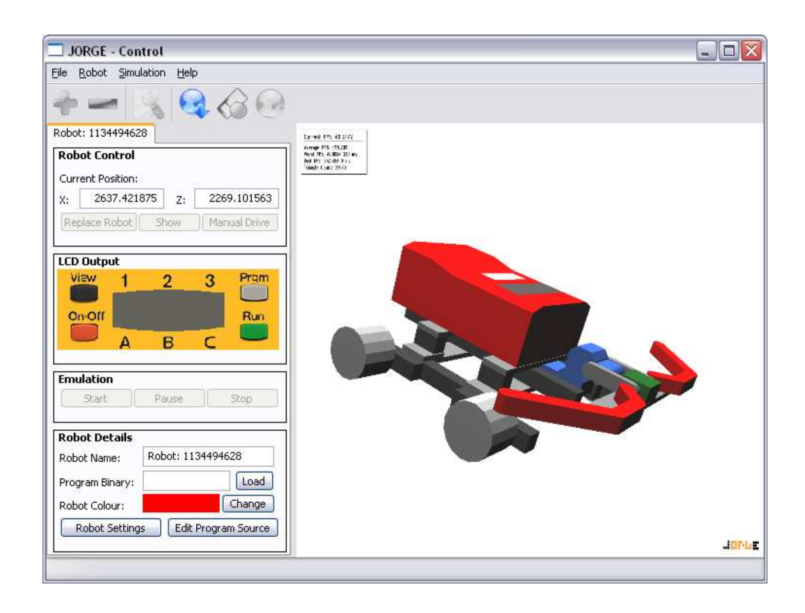



Abbildung 1.6: Roboter Ansehen

In diesem Modus sind folgende Aktionen möglich:

- Mit den Pfeilen der Tastatur kann der Roboter in jede Richtung gedreht werden.

## 1.7 Hauptfenster - Simulations Modus

Nachdem Sie vom *Roboter Ansehen Modus* zurück in den *Welt Konfigurieren Modus* gewechselt haben (Menu Simulation ▷ Prepare Simulation), kann die Simulation nun gestartet werden:

Lösen Sie dazu die Aktion Simulation ▷ Start Simulation aus. Sobald der *Simulations Modus* aktiviert ist, werden alle Roboter bzw. dessen Programme gestartet. Im *Robot Log* werden alle Instruktionen, mit denen das Programm den virtuellen Roboter steuert, gezeigt. Beim Beispiel aus dem Abschnitt 1.5.1 wird folgendes Ausgegeben:

```
Reading binary: C:\java\HelloRobot.emu
Installing binary 80240944
Checking validity of magic number
Engine starting.
Set Motor 0 Forward
Set Motor 2 Forward
Set Motor 2 Brake
Set Motor 2 Brake
Set Motor 0 Brake
Set Motor 0 Backward
Set Motor 2 Forward
Set Motor 0 Brake
Set Motor 2 Brake
Reading binary: C:\java\HelloRobot.emu
Installing binary 80140032
Checking validity of magic number
Engine finished.
```

Das Programm kann nun beliebig oft über das *RobotTab* gestartet, pausiert oder beendet werden. Wenn Sie das Programm nun verändern möchten, können Sie dies tun, indem Sie zurück in den *Welt Konfigurieren Modus* wechseln (Menü Simulation ▷ Prepare Simulation).

# Kapitel 2

# Überblick

In diesem Kapitel werden alle Komponenten der Benutzeroberfläche vorgestellt und erklärt.

## 2.1 Das Hauptfenster

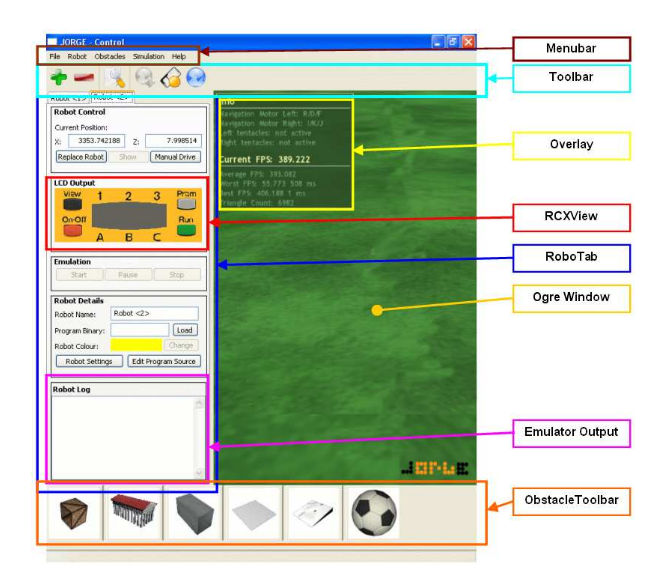



Abbildung 2.1: Hauptfenster Übersicht

Die obige Abbildung zeigt alle Komponenten die in den folgenden Abschnitten erläutert werden.

### 2.1.1 Menüleiste

Jedes Menü in der Menüleiste beinhaltet Aktionen, die sich auf eine Simulation beziehen. Funktionen die man oft benötigt sind auch über die Toolbar zu erreichen.

- *File*
  - *New Simulation:* Erstellt eine neue Simulation, öffnet den Dialog 2.2.2
  - *Open Simulation:* Öffnet eine Simulation, die in eine Datei abgespeichert wurde.
  - *Close Simulation:* Schliesst eine bereits geöffnete Simulation.
  - *Save:* Speichert eine veränderte Simulation ab.

  - *Save As:* Speichert eine Simulation in einer Datei.
  - *Settings:* Öffnet den Dialog Einstellungen 2.2.2.
  - *Exit:* Beendet JORGE

- *Robot*
  - *Add Robot:* Fügt einer bereits geöffneten Simulation ein Roboter hinzu. Für jeden Roboter wird ein neuer *RobotTab* erstellt 2.1.3
  - *Remove Robot:* Entfernt den zum ausgewählten *RobotTab* gehörenden Roboter aus der Simulation.
  - *View Robot:* Aktiviert den Modus: *Roboter Ansehen* 2.3.3

- *Simulation*
  - *World Settings:* Öffnet den Dialog: 2.2.2
  - *Prepare Simulation:* Wechselt in den Modus *Welt Konfigurieren Modus* 2.3.2
  - *Start Simulation:* Wechselt in den Modus *Simulation 2.3.4*

- *Help*
  - *Open Documentation:* Öffnet diese Dokumentation.
  - *About:* Informationen über die Entwickler von JORGE.

### 2.1.2 Toolbar

Die Toolbar enthält häufig gebrauchte Aktionen, die auch über die Menüleiste erreichbar sind. Die Schaltflächen der Menüleiste sind nicht mit Text bezeichnet. Halten Sie die Maus ein paar Sekunden über die Schaltflächen, damit eine Beschreibung angezeigt wird. Die sich jeweils entsprechenden Aktionen aus der Menüleiste und der Toolbar besitzen das gleiche Icon.

### 2.1.3 RobotTab

Ein *RobotTab* repräsentiert einen Roboter. Die einzelnen Bestandteile werden in diesem Abschnitt beschrieben, und zwar in der Reihenfolge wie sie auch auf der Benutzeroberfläche von oben nach unten auftreten.

#### 2.1.3.1 Robot Control

Hier wird die Start Position (*Current Position*) des Roboters angezeigt. Die Start Position legt fest, von wo der Roboter startet wenn Sie die Simulation starten. Nutzen Sie die Schaltfläche **Replace Robot**, um den Roboter an den Ort zu verschieben, der durch das rote Kreuz markiert wird.

**Manual Drive** ermöglicht es Ihnen, den Roboter von Hand zu steuern. Folgende Tasten der Tastatur können zur Navigation genutzt werden:

- *R:* Linker Motor vorwärts
- *I:* Rechter Motor vorwärts
- *D:* Linker Motor stop
- *K:* Rechter Motor stop
- *F:* Linker Motor rückwärts
- *J:* Rechter Motor rückwärts

*Bemerkung:* Diese Tastatureingaben sind nur möglich wenn sich der Mauszeiger innerhalb des 3D Window (siehe 2.1.4) befindet.

#### 2.1.3.2 LCD Output

Das Roboter Programm kann hier Zahlen und Symbole anzeigen. LeJOS sieht vor, dass auf dem LCD Display auch Buchstaben ausgegeben werden können, dies wir von JORGE nicht unterstützt. Diese Anzeige hat experimentellen Status, d.h. die Entwickler von JORGE können nicht garantieren, dass die Ausgabe korrekt ist.

#### 2.1.3.3 Emulation

Über die Schaltflächen **Start**, **Pause** und **Stop** kann im Simulations Modus (siehe Abschnitt 2.3.4) das Roboter-Programm kontrolliert werden.

#### 2.1.3.4 Robot Details

Jedem Roboter kann ein **Name** vergeben werden. Um den Namen dauerhaft zu ändern, schliessen Sie die Eingabe des Namens mit der Enter-Taste ab, der Angezeigte Name im Reiter des *RoboTabs* wird nun aktualisiert.

Mit der Schaltfläche **Load** kann ein bereits kompiliertes und gelinktes Roboter-Programm in den virtuellen Roboter geladen werden. Wenn Sie das Programm mit JORGE erstellt haben, trägt die zu ladende Datei die Endung *.emu*. Wenn ein Programm geladen ist, wird dessen Dateiname im Textfeld **Program Binary** angezeigt. Sie können ein bereits geladenes Programm auch ersetzen.

Im Modus *Roboter Ansehen* könne Sie die Farbe des Roboters ändern. Dazu klicken Sie auf die Schaltfläche **Change** und wählen im Dialog die gewünschte Farbe aus.

Die Schaltfläche **Robot Settings** öffnet den Dialog *Robot Settings* (siehe Abschnitt 2.2.3) und die **Edit Program Source** startet den Editor (Abschnitt 2.6).

#### 2.1.3.5 Robot Log

Wenn das Roboter-Programm Instruktionen an den virtuellen Roboter schickt, werden die Details in diesem Textfeld ausgegeben.

### 2.1.4 3D Window

Dieser Teil der Benutzeroberfläche zeigt die virtuelle Welt an. Je nach Modus (siehe Kapitel 2.3) nimmt die Maus und die Tastatur eine andere Funktion an.

#### 2.1.4.1 Overlay

Hier sind Einzelheiten der Grafikausgabe angezeigt:

- *Current FPS:* Diese Zahl gibt an, wie oft der Inhalt des 3D Windows neu gezeichnet wird (pro Sekunde).
- Average FPS: Die durchschnittliche Anzahl gezeichneter Frames pro Sekunde.
- *Best FPS:* Der Beste je erreichte Wert, seit Sie JORGE gestartet haben.
- *Triangle Count:* Die Anzahl Dreiecke aus der das angezeigte Bild aufgebaut ist.

### 2.1.5 Obstacle Toolbar

Die Obstacle Toolbar zeigt Hindernisse an, die in die Simulation eingefügt werden können. Wenn Sie auf das entsprechende Symbol klicken, wird das Hindernisse an dem Ort eingefügt, der mit dem roten Kreuz markiert ist. Wie Sie selber Hindernisse zu JORGE hinzufügen, ist im Kapitel 3.1 beschrieben.

## 2.2 Dialoge

### 2.2.1 Einstellungen

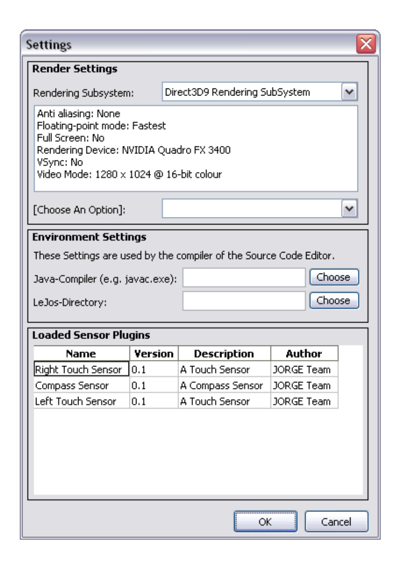



Abbildung 2.2: Einstellungen

Dieser Dialog fasst alle Einstellungen die JORGE betreffen zusammen.

#### 2.2.1.1 Render Settings

Die **Render Settings** definieren das 3D Ausgabesystem, JORGE unterstützt derzeit nur *DirectX 9.* Wenn Sie Schwierigkeiten mit der Anzeige haben, sollten Sie die Einstellungen für das **Rendering Device** überprüfen: Dazu aktivieren Sie die entsprechende Zeile im Textfeld und wählen Ihre Grafikkarte in der Liste aus(in Abbildung 2.2 ist die Liste mit *[Choose An Option]* angeschrieben).

### 2.2.2 World Settings

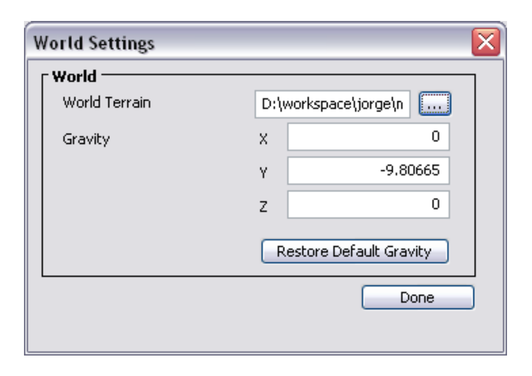



Abbildung 2.3: Welt Einstellungen

Der Dialog **World Settings** fasst Einstellungen der virtuellen Welt zusammen. Mit der Option **World Terrain** können Sie eine Datei angeben, die festlegt, wie die virtuelle Welt aussieht (welche Voraussetzungen erfüllt sein müssen, können Sie dem Abschnitt 3.2 entnehmen).

In den Textfeldern, die mit **Gravity** zusammengefasst werden, legen Sie die Richtung und die Beschleunigung ($\frac{m}{s^2}$) der Erdanziehung fest. Abbildung 2.4 zeigt die Ausrichtung des Koordinatensystems.

Bitte beachten Sie, dass Sie mit **X,Y,Z** einen Richtungsvektor beschreiben. Wenn also die Gravitation senkrecht nach unten zeigen soll, muss die Y-Komponente negativ sein.

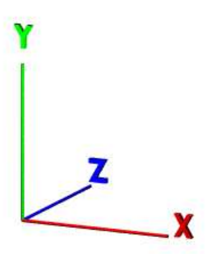



Abbildung 2.4: Koordinatensystem

### 2.2.3 Robot Settings

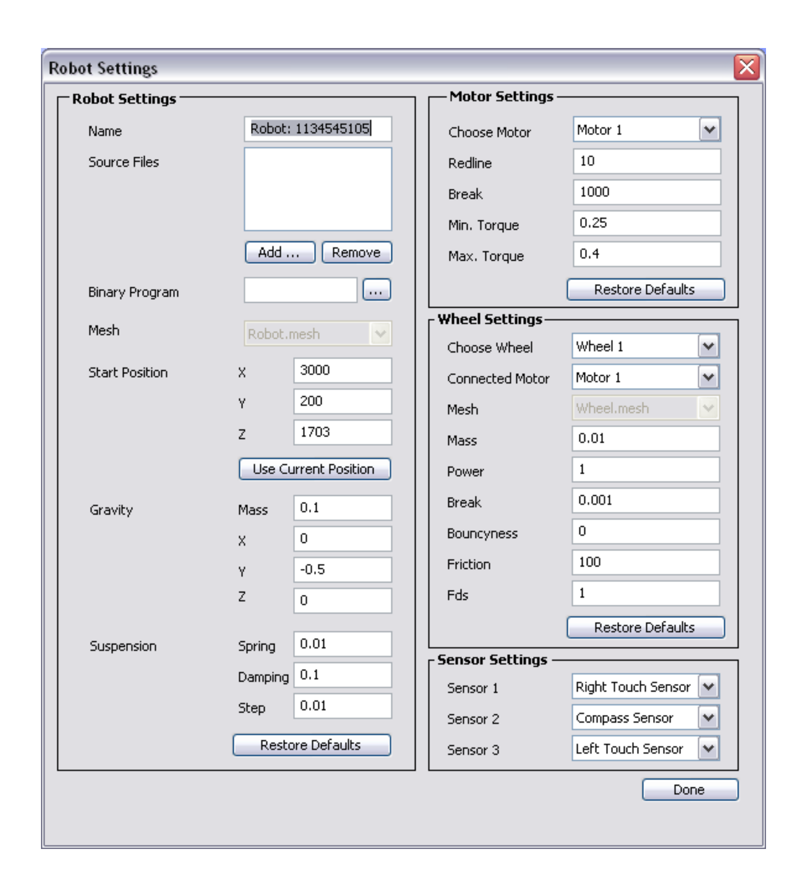



Abbildung 2.5: Roboter Einstellungen

Das Fenster **Robot Settings** fasst alle Einstellungen eines Roboters zusammen. Sie werden in der JORGE-Projektdatei abgespeichert, so dass eine Simulation jederzeit wieder hergestellt werden kann.

#### 2.2.3.1 Robot Settings

- **Name**: Ein frei wählbarer Name mit dem Sie einen einzelnen Roboter benennen können. Der Name wird im Reiter des *RobotTab*s angezeigt.

- **Source Files**: Hier können Sie die Quelltexte, die zu einem Roboter-Programm gehören, einem Roboter zuordnen. Sie werden automatisch im *Editor* geöffnet werden, wenn Sie im *RobotTab* auf die Schaltfläche *Edit Program Source* klicken.

- **Start Position**: Beschreibt die Ausgangsposition des Roboters in der virtuellen Welt. Achten Sie darauf, dass die gewählten Koordinaten vernünftige Werte annehmen, sonst verschwindet der Roboter von der Bildfläche (die maximalen Werte sind abhängig vom ausgewählten Terrain).

- **Gravity**: Hier können Sie den Massenmittelpunkt und die physikalische Masse des Roboters festlegen.

  - **Mass**: Das Gewicht des Roboters in Kilogramm.
  - **X**,**Y** und **Z:** Beschreibt den Massenmittelpunkt des Roboters. Die Koordinate $\begin{matrix}0\\0\\0\end{matrix}$ beschreibt genau die Mitte des Robotermodells.

- **Suspension**: Beschreibt die Federung der Roboter-Räder. Um den Rechenaufwand nicht unnötig zu erhöhen, haben alle Räder die gleichen Feder-Eigenschaften.

  - **Spring**: Die Federkonstate der Federung (eine kleine Zahl beschreibt eine weiche Federung).
  - **Damping**: Dies ist ein Mass für die Dämpfung der Federung (wie lange es geht bis die Schwingung aufhört).
  - **Step**: Eine Konstante zur Berechnung der Federung.

- **Restore Defaults**: Setzt die Einstellungen auf die Vorgabe zurück.

#### 2.2.3.2 Motor Settings

- **Choose Motor**: Hier können Sie den Motor auswählen, dessen Einstellungen Sie verändern möchten.

- **Redline**: Die maximale Drehzahl des Motors.

- **Break**: Die Bremskraft des Motors.

- **Min. Torque**: Das minimale Drehmoment des Motors.

- **Max. Torque**: Das maximale Drehmoment des Motors.

- **Restore Defaults**: Setzt die Einstellungen auf die Vorgabe zurück.

#### 2.2.3.3 Wheel Settings

- **Choose Wheel**: Hier können Sie das Rad auswählen, dessen Einstellungen Sie verändern möchten.

- **Connected Motor**: Bestimmt welcher Motor dieses Rad antreibt.

- **Mesh**: Diese Option ist deaktiviert, mit ihr könnte ein eigenes 3D Modell für das Rad gewählt werden.

- **Mass**: Die Masse des Rades in Kilogramm.

- **Power**: Ist ein Mass für die Kraftübertragung vom Motor zum Rad, und entspricht einer Fliehkraftkupplung.

- **Break**: Wie stark die Räder von der Achse gebremst werden können.

- **Bouncyness**: Ein Mass für den Luftdruck im Rad oder wie fest der Roboter hüpft.

- **Friction**: Die ist ein Mass für die Reibung die zwischen dem Boden und den Rädern entsteht.
- **Fds**: Steht für "Force Dependent Slips", also wie fest die Räder bei hohen Kräften zum Rutschen neigen.
- **Restore Defaults**: Setzt die Einstellungen auf die Vorgabe zurück.

#### 2.2.3.4 Sensor Settings

Hier definieren Sie welche Sensoren an den Roboter angeschlossen sind. JORGE kann dank Plugins mit Sensoren erweitert werden (siehe Abschnitt 3.3). Da der RCX nur drei Sensor-Eingänge besitzt, müssen Sie hier festlegen, welche Sensoren eingesetzt werden.

- **Sensor 1**: LeJOS: *Sensor.S1*
- **Sensor 2**: LeJOS: *Sensor.S2*
- **Sensor 3**: LeJOS: *Sensor.S3*

### 2.2.4 Editor

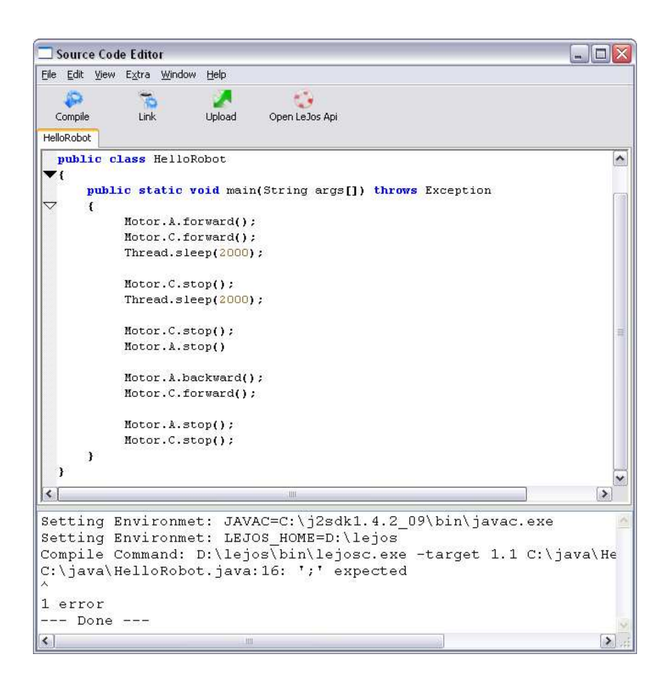



Abbildung 2.6: Editor

Mit dem Editor können Programme für die Roboter erstellt werden. Der Editor bietet sehr viele Funktionen die Sie auch in jedem anderen Texteditor antreffen. Daher werden hier nur die Eigenheiten dieses Editors beschrieben:

- **Compile**: Kompiliert den aktuellen Quelltext (aktiver Reiter im Editor) und alle Klassen, die referenziert werden. Wenn Sie den Quelltext selber in einer Konsole kompilieren wollen, können Sie die nötigen Befehle dem Ausgabefenster entnehmen.
- **Link**: Der Java-Compiler erstellt für jede Klasse eine *.class* Datei. Diese Dateien müssen zu einer Datei (mit der Endung *.emu*) zusammengefügt werden, damit man sie auf den Roboter laden kann. Beim Linken werden ausserdem die eingesetzten LeJOS Klassen zur resultierenden Datei hinzugefügt. Wenn Ihr Programm aus mehreren Java-Dateien besteht, müssen Sie im Editor diejenige Datei aktivieren, welche die *main* Methode enthält, damit der Linker-Vorgang erfolgreich sein kann.

- **Upload**: Lädt das gelinkte Programm in den Roboter. Wie beim linken, muss im Editor die Datei ausgewählt sein, welche die *main* Methode enthält. Programme können nur im Modus *Welt Konfigurieren* in den Roboter geladen werden.
- **Open LeJos Api**: Öffnet das LeJOS API (Java Dokumentation) in einem externen Browser.

## 2.3 Modi

JORGE unterscheidet vier verschiedene Modi, die in diesem Abschnitt erklärt werden. Die Toolbar zeigt an, in welchem Modus Sie sich gerade befinden.

### 2.3.1 Modus: Simulation geschlossen



Abbildung 2.7: Simulation Geschlossen

Wenn Sie JORGE starten, ist JORGE solange in diesem Modus, bis Sie eine Simulation durch die entsprechende Aktion im Menü File erstellen oder öffnen. Wenn Sie eine geöffnete Simulation über das Menü File schliessen gelangen Sie wieder in diesen Zustand.

### 2.3.2 Modus: Welt Konfigurieren



Abbildung 2.8: Welt Konfigurieren

Sobald eine Simulation geöffnet wird, wechselt JORGE in diesen Modus. Hier geht es darum, dass Sie die Roboter für die Simulation vorbereiten, d.h. Sie haben die Möglichkeit Roboter und Hindernisse in die Welt einzufügen und die gewünschten Einstellungen vorzunehmen. In diesem Modus stehen folgende Aktionen zur Verfügung:

- Die 3D Welt erkunden, siehe Abschnitt 1.3
- Hindernisse einfügen, siehe Abschnitt 1.3.1
- Roboter einfügen, siehe Abschnitt 1.3.2
- Roboter konfigurieren, siehe Abschnitt 2.1.3 und 2.2.3
- Den Roboter manuell steuern, siehe Abschnitt 2.1.3.1
- Bestehendes Roboter Programm laden, siehe Abschnitt 1.4
- Roboter Programm erstellen, siehe Abschnitt 1.5.1
- In den *Roboter Ansehen Modus* wechseln
- In den *Simulations Modus* wechseln

### 2.3.3 Modus: Roboter Ansehen



Abbildung 2.9: Roboter Ansehen

In diesem Modus können Sie den Roboter ansehen und dessen Farbe verändern. Sie können

- die Farbe des Roboters ändern, siehe Abschnitt 2.1.3.4.
- den Roboter ansehen, siehe Abschnitt 1.6
- zurück in den Modus *Welt Konfigurieren* wechseln.

### 2.3.4 Modus: Simulation



Abbildung 2.10: Simulation

Beim Wechsel in diesen Modus werden alle Roboter-Programme gestartet. Sie können:

- die Kamera bewegen, wie in Abschnitt 1.3 beschrieben.
- Roboter stoppen, starten oder pausieren, siehe Abschnitt 2.1.3.3
- zurück in den Modus *Welt Konfigurieren* wechseln.

# Kapitel 3

# JORGE Erweitern

## 3.1 Neue Hindernisse Erstellen

Es ist möglich eigene Hindernisse zu definieren, und sie zu JORGE hinzuzufügen. Dazu müssen Sie im Unterverzeichnis *<JORGE Installations-Ordner>/media/obstacles/* vier neue Dateien anlegen (Ersetzen Sie <name> mit einem Wort Ihrer Wahl):

- *<name>.xml* definiert die Masse sowie die Form der Bounding-Box, welche benutzt werden soll: *box* oder *sphere. box* und *sphere* sind dynamische Hindernisse, die in der 3D Szene verschoben werden können. Es können auch Hindernisse als *static* definiert werden. Das bedeutet, dass keine Bounding-Box sondern jedes einzelne Polygon kollidiert. Dabei ist zu beachten, dass die Position eines *static* Hindernisses unveränderbar ist.

```
<?xml version="1.0" encoding="utf-8"?>
<obstacle xmlns="http://jorge.hta-bi.ch/jorgeobstacle"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://jorge.hta-bi.ch/jorgeobstacle
                            file:./jorge-obstacle.xsd">
       <mass>100</mass>
       <type>box</type>
</obstacle>
```

- *.<name>.mesh* die Daten des 3D Modells: Diese Datei muss mit einem 3D Modellierer erstellt werden. Eine gute Anleitung finden Sie

- *<name>.material* legt die Oberflächen Textur fest, ein Beispiel:

```
material Ac3d/Crate/Mat001_Tex00
{
   technique
   {
     pass
     {
         ambient 1 1 1 1
         diffuse 1 1 1 1
         texture_unit
         {
             texture crate.png
         }
     }
   }
}
```

Eine ausführliche Anleitung finden Sie auf der OGRE `http://www.ogre3d.org/docs/manual/manual_15.html` Homepage.

- <name>.gif Eine Bilddatei (die Formate bmp, gif, xbm und xpm werden unterstützt), die wenn möglich 80x80 Pixel gross ist und als Vorschau auf der Benutzeroberfläche in der ObstacleToolbar angezeigt wird.

## 3.2 Neues Terrain Erstellen

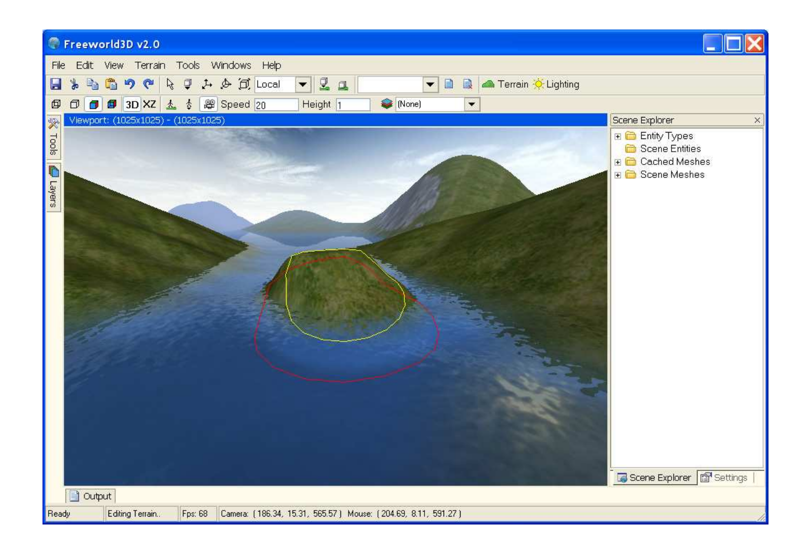



Abbildung 3.1: Screenshot Freeworld3D

Das Terrain kann über mehrere Wege erstellt werden. In diesem Abschnitt werden allgemeine Informationen zur Erstellung eines Terrains gegeben.

Das Rendering eines Terrains wird in OGRE von der Klasse TerrainSceneManager übernommen. Für die korrekte Interpretation eines Terrains sind folgende Dateien notwendig:

- Heightmap - ist ein Graustufen-Bild, in welchem jedes Pixel einen Höhenwert repräsentiert. Der Farbwert 0 steht für den tiefsten Punkt, der Farbwert 255 für den Höchsten.
- Terrain Textur - bildet das Terrain von oben ab.
- Detail Terrain Textur - wird verwendet, wenn der Betrachter das Terrain von nahem betrachtet.
- *.cfg - OGRE spezifisch. Trägt die Daten zusammen und enthält zusätzliche Informationen wie die Bemessung des Terrains oder die maximale Höhe im Koordinatensystem von OGRE.

Weitere Informationen, und die genauen Angaben zur *.cfg Datei finden Sie unter Terrain howto `http://www.ogre3d.org/wiki/index.php/Terrain_Scene_Manager`.

Abfolge der Schritte für die Erstellung eines Terrains:

1. Erstellung der Heightmap mit Hilfe eines Tools wie Freeworld3D oder Terragen
2. Texture mit Hilfe der oben genannten Tools darüber legen
3. Export der Heightmap in eine *.bmp oder *.raw Datei
4. Export der Terrain Textur und der Detail Terrain Textur
5. Erstellung der *.cfg Datei, eine Vorlage kann im Ordner <*JORGE Installations-Ordner*>*/media/config* gefunden werden

## 3.3 Sensoren hinzufügen

JORGE wurde so konzipiert, dass andere Entwickler neue Sensoren erstellen können, und Sie diese Sensoren zu JORGE hinzufügen können. Damit JORGE diese neuen Sensoren lädt, müssen Sie die Dateien, die Sie vom Entwickler (oder von dessen Website) erhalten, in das Verzeichnis *<JORGE Installations-Verzeichnis>/media/plugins/Release/* kopieren und JORGE gegebenenfalls neu laden. Der Dialog Einstellungen (siehe 2.2.1) zeigt alle geladenen Sensoren an.

# Anhang A

# Java Installation

Eine Java Installation ist nicht nötig, wenn Sie vorkompilierte Roboter-Programme ausführen wollen. Wenn Sie jedoch Programme selber schreiben, ist ein Java SDK (Software Development Kit) unerlässlich. Java kann kostenlos von *http://java.sun.com* heruntergeladen werden. Dort finden Sie auch ausführliche Installationsanleitungen.

Merken Sie sich bei der Installation das Verzeichnis, in welches Sie Java installieren. Dieses Verzeichnis hat normalerweise einen Unterordner *bin/*, der *das* Programm *javac.exe* enthält.

Für dieses Handbuch wurden folgende Konfiguration verwendet:

- Java Version: *j2sdk1.4.2_09*
- Installationsverzeichnis: *C:*

Weitere Informationen: http://java.sun.com/j2se/1.4.2/install-windows.html `http://java.sun.com/j2se/1.4.2/install-windows.html`

# Anhang B

# LeJOS Installation

LeJOS kann von *http://lejos.sf.net* heruntergeladen werden. Für LeJOS gibt es kein Installations Programm: Es reicht aus, die Datei (*lejos_win32_2_1_0.zip*) in ein Verzeichnis Ihrer Wahl zu entpacken. Merken Sie sich wo Sie LeJOS entpackt haben, da Sie dieses Verzeichnis JORGE angeben müssen wenn Sie Programme kompilieren wollen.

Für dieses Handbuch wurden folgende Konfiguration verwendet:

- LeJOS Version: *2.1.0 (Windows Version)*
- Installationsverzeichnis: *D:\lejos*

Anmerkung: LeJOS setzt eine Java Installation voraus.

Weitere Informationen: http://lejos.sourceforge.net/installwindows.html `http://lejos.sourceforge.net/installwindows.html`

# Index